## Engenharia da Computação

## Programação de Hardware

**2º Semestre 2022**

## Eduardo Guimarães Corrêa – 185698

## Henrico Santa Rosa – 186099

## Luiz Ricardo Fávaro - 185675

## Manoela Alvares de Abreu Oliveira – 190565

1. **Projeto Carrinho seguidor de linha**

# Componentes

* + - Arduino Nano;
    - 3x Sensor de obstáculo infravermelho – 3,3V a 5V;
    - 2x TIP41C;
    - 3x Borne;
    - Barra de pino;
    - Diodo Schottky 1N5822 – 40V 3A;
    - 2x Resistores – 220 Ohms;
    - 2x Resistores – 47 Ohms;
    - 2x Motor N20 – 6v 1000rpm
    - 2x Baterias lithium – 1100mah 3.7V;
    - Jumper;
    - Filamento PLA;

# Circuito

# Esquemático:

# Diagrama, Esquemático Descrição gerada automaticamente

# PCB:

# Interface gráfica do usuário Descrição gerada automaticamente

# 3D:

# Circuito eletrônico em fundo preto Descrição gerada automaticamente

# Código Arduino

# // DEFINIÇÕES DE PINOS #define pinSensorEsq A0 #define pinSensorMeio A1 #define pinSensorDir A2 #define motorEsq 5 #define motorDir 6 // Declaração de Funções void configMotor(); // Declaração de variáveis bool leituraEsquerda; bool leituraMeio; bool leituraDireita; void setup() {   pinMode(pinSensorMeio, INPUT);   pinMode(pinSensorDir, INPUT);   pinMode(pinSensorEsq, INPUT);   pinMode(motorEsq, OUTPUT);   pinMode(motorDir, OUTPUT); } void loop() {   if (analogRead(pinSensorEsq) <= 500 && analogRead(pinSensorDir) <= 500 && analogRead(pinSensorMeio) >= 500)   {     analogWrite(motorEsq, 200);     analogWrite(motorDir, 200);   }   else if (analogRead(pinSensorEsq) >= 500)   {     analogWrite(motorEsq, 40);     analogWrite(motorDir, 150);     delay(175);   }   else if (analogRead(pinSensorDir) >= 500)   {     analogWrite(motorEsq, 150);     analogWrite(motorDir, 40);     delay(175);   } }

# Código em C

/\*  Projeto Carro Seguidor de Linha  
 \*    
 \*  Integrantes:  
 \*    
 \*  Eduardo Guimarães | RA: 185698  
 \*  Henrico dos Santos | RA: 186099  
 \*  Luiz Ricardo | RA: 185675  
 \*  Manoela Alvares | RA: 190565  
 \*    
 \*/  
  
#define F\_CPU 16000000  
  
#include <avr/io.h>  
#include <util/delay.h>  
  
#define motorEsq 5 //Definição da porta utilizada para o motor esquerdo - D5  
#define motorDir 6 //Definição da porta utilizada para o motor direito - D6  
  
int sensorEsq = 0; //Definição da variável que vai ler a porta analógica do sensor esquerdo - A0  
int sensorMeio = 0; //Definição da variável que vai ler a porta analógica do sensor do meio - A1  
int sensorDir = 0; //Definição da variável que vai ler a porta analógica do sensor direito - A2  
  
void InitADC(){  
  ADMUX |= (1<<REFS0);  
  ADCSRA |= (1<<ADPS2) | (1<<ADPS1) | (1<<ADPS0) | (1<<ADEN); //Inicialização do ADC e configuração do prescaler de 128  
}  
  
uint16\_t ReadADC(uint8\_t ADCchannel){  
  ADMUX = (ADMUX & 0xF0) | (ADCchannel & 0x0F); //Seleção do canal ADC  
  ADCSRA |= (1<<ADSC); //Modo: Single Mode  
  while(ADCSRA & (1<<ADSC)); //Loop para esperar até a conversão de ADC ser realizada  
  return ADC;  
}  
  
void motorEsqPWM(){  
  TCCR0A |= (1 << COM0A1)  | (0 << COM0A0) | (1 << WGM01) | (1 << WGM00); //Configuração do PWM para o motor esquerdo  
  TCCR0B |= (0 << CS02) | (0 << CS01) | (1 << CS00); //Sem uso de prescaler  
}  
  
void motorDirPWM(){  
  TCCR0A |= (1 << COM0B1)  | (0 << COM0B0) | (1 << WGM01) | (1 << WGM00); //Configuração do PWM para o motor direito  
  TCCR0B |= (0 << CS02) | (0 << CS01) | (1 << CS00); //Sem uso de prescaler  
}  
  
int leituraSensor(){  
  //OCR0A: Motor Esquerdo | OCR0B: Motor Direito | Duty Cycle  
  sensorEsq = ReadADC(0); //Faz a leitura do sensor esquerdo conectado na porta A0  
  sensorMeio = ReadADC(1); //Faz a leitura do sensor do meio conectado na porta A1  
  sensorDir = ReadADC(2); //Faz a leitura do sensor direito conectado  na porta A2  
  if((sensorEsq >= 500) && (sensorMeio < 500) && (sensorDir < 500)){ //Sensor Esq: preto | Sensor Meio: branco | Sensor Dir: branco (curva para esquerda para corrigir)  
    OCR0A = 5;  
    OCR0B = 80;  
    \_delay\_ms(175);  
  }  
  if((sensorEsq < 500) && (sensorMeio >= 500) && (sensorDir < 500)){ //Sensor Esq: branco | Sensor Meio: preto | Sensor Dir: branco (linha reta)  
    OCR0A = 80;  
    OCR0B = 80;  
  }  
  if((sensorEsq < 500) && (sensorMeio < 500) && (sensorDir >= 500)){ //Sensor Esq: branco | Sensor Meio: branco | Sensor Dir: preto (curva para direita para corrigir)  
    OCR0A = 80;  
    OCR0B = 5;  
    \_delay\_ms(175);  
  }  
  if((sensorEsq >= 500) && (sensorMeio >= 500) && (sensorDir < 500)){ //Sensor Esq: preto | Sensor Meio: preto | Sensor Dir: branco (curva para esquerda)  
    OCR0A = 5;  
    OCR0B = 80;  
    \_delay\_ms(175);  
  }  
  if((sensorEsq < 500) && (sensorMeio >= 500) && (sensorDir >= 500)){ //Sensor Esq: branco | Sensor Meio: preto | Sensor Dir: preto (curva para direita)  
    OCR0A = 80;  
    OCR0B = 5;  
    \_delay\_ms(175);  
  }  
  if((sensorEsq >= 500) && (sensorMeio >= 500) && (sensorDir >= 500)){ //Sensor Esq: branco | Sensor Meio: branco | Sensor Dir: preto (menor probabilidade dos 3 sensores identificarem a fita preta)  
    OCR0A = 80;  
    OCR0B = 80;  
    \_delay\_ms(175);  
  }  
}  
  
int main(void){  
  DDRD |= (1 << motorEsq) | (1 << motorDir); //Definição das portas direcionadas ao uso dos motores  
  InitADC(); //Chamada da função que inicializa o ADC  
  motorEsqPWM(); //Chamada da função que configura o PWM do motor esquerdo  
  motorDirPWM(); //Chamada da função que configura o PWM do motor direito  
  while(1){  
    leituraSensor(); //Chamada da função lê os valores de input dos pinos analógicos para então poder configurar o PWM de cada um dos motores  
  }

# }

# Desing 3D e Projeto Final

# 3D:

# Em preto e branco Descrição gerada automaticamente com confiança média

# Projeto Final:

# Imagem digital fictícia de personagem de desenho animado Descrição gerada automaticamente com confiança baixa

# Imagem de jogo de vídeo game Descrição gerada automaticamente com confiança baixa

# Uma imagem contendo bolo, mesa, aniversário Descrição gerada automaticamente